



Todas las respuestas se consideran válidas **solo** si están debidamente justificadas.

## Ejercicio 1

Se tiene un enlace de 1Mbps y sobre el mismo se implementa un protocolo de máxima eficiencia que utiliza tanto reconocimiento selectivo y acumulativo. El largo de los frames utilizado es de 2kb.

- a. Calcule el delay que deben tener los frames para que la ventana del emisor sea igual a 256 frames.

Rta:

Dado que la eficiencia es máxima, el delay se puede calcular con:

$$SWS = V_{tx} \cdot RTT / |Frame|$$

$$Delay = \frac{1}{2} SWS \cdot |Frame| / V_{tx} = \frac{1}{2} 256 \cdot 2kb / 1000kbps = 0.256 segundos$$

- b. Usando el mismo tamaño de ventana de emisión, detalle gráficamente, incluyendo el tamaño de cada campo, los frames de emisión y recepción. Tener en cuenta que el protocolo usa piggybacking.

Rta:

Los campos de seq, ack y sack deben ocupar  $\lceil \log_2(SWS + RWS) \rceil bits = \lceil \log_2(512) \rceil bits = \lceil 9 \rceil bits = 9bits$

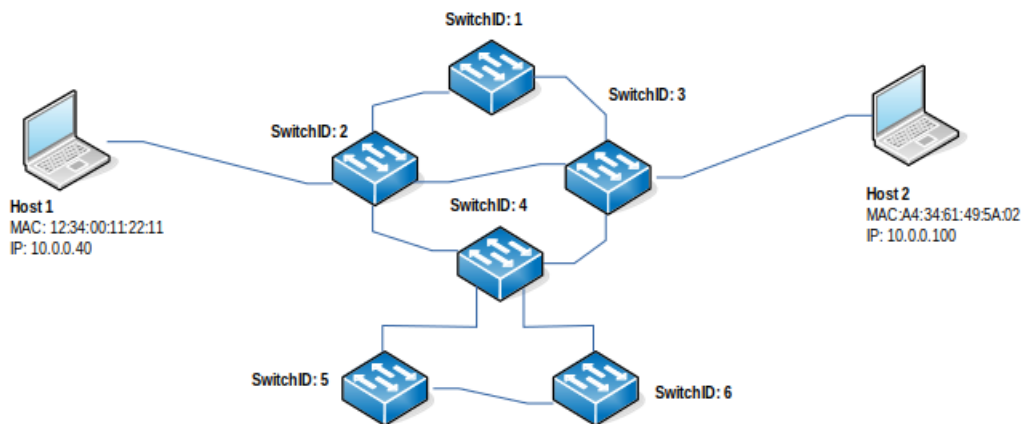
Como usa piggybacking, el emisor y el receptor usan el mismo frame. Uso un CRC de 16 bits para controlar errores

Frame : |SEQ(9bits)|ACK(9bits)|SACK(9bits)|Datos|CRC(16bits)|

## Ejercicio 2

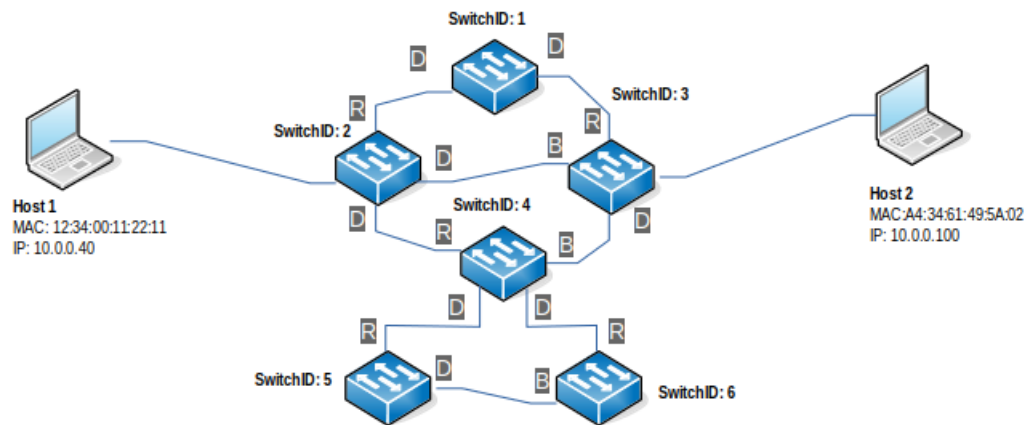
Dada la red de la figura, en la cual el protocolo STP ya ha convergido y las tablas de forwarding de los switchs se encuentran vacías, se pide:

- a. Si el Host 1 envía un mensaje ARP request a la red y el Host 2 contesta con un ARP reply: ¿Qué entradas se aprenden una vez terminado el intercambio de mensajes?



Rta:

La red después de STP queda como en la figura:



Después de enviar el ARP request:

Switch 1 aprende que H1 está en la interface de la izquierda  
 Switch 2 aprende que H1 está en la interface de la izquierda  
 Switch 3 aprende que H1 está en la interface de arriba  
 Switch 4 aprende que H1 está en la interface de la izquierda  
 Switch 5 aprende que H1 está en la interface de arriba  
 Switch 6 aprende que H1 está en la interface de arriba

Después de enviar el ARP reply:

Switch 1 aprende que H2 está en la interface de la derecha  
 Switch 2 aprende que H2 está en la interface de arriba  
 Switch 3 aprende que H2 está en la interface de la derecha

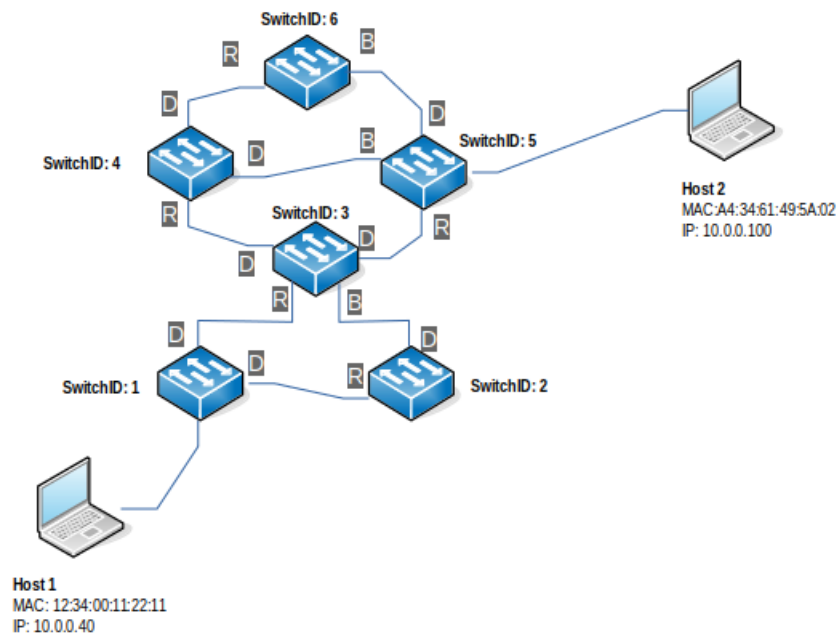
b. Si luego el Host 1 se conecta al switch 5 y envía un mensaje unicast al Host 2 (las tablas contienen lo aprendido). Indique:

- Cómo quedan las mismas una vez terminado el intercambio de mensajes.
- Reconfigure la red tal que el delay entre los hosts 1 y 2 disminuya con respecto a la configuración dada en el punto anterior (Host 1 conectado a switch 5). Suponga que el delay de cada enlace es constante y el mismo. Indique con una figura en qué estado queda cada uno de los puertos y el Id de cada uno de los switches para la configuración final.

Rta: Después del envío del mensaje:

Switch 1 renueva que H1 está en la interface de la izquierda  
 Switch 2 reaprende que H1 está en la interface de abajo  
 Switch 3 renueva que H1 está en la interface de arriba  
 Switch 4 reaprende que H1 está en la interface de abajo a la izquierda  
 Switch 5 reaprende que H1 está en la interface de la izquierda  
 Switch 6 queda igual

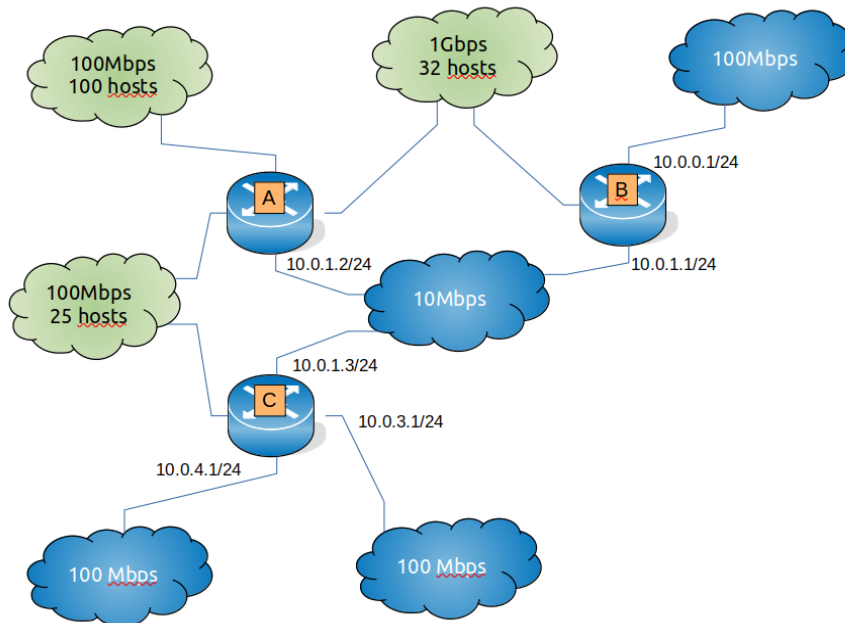
Una configuración posible para minimizar el delay:



### Ejercicio 3

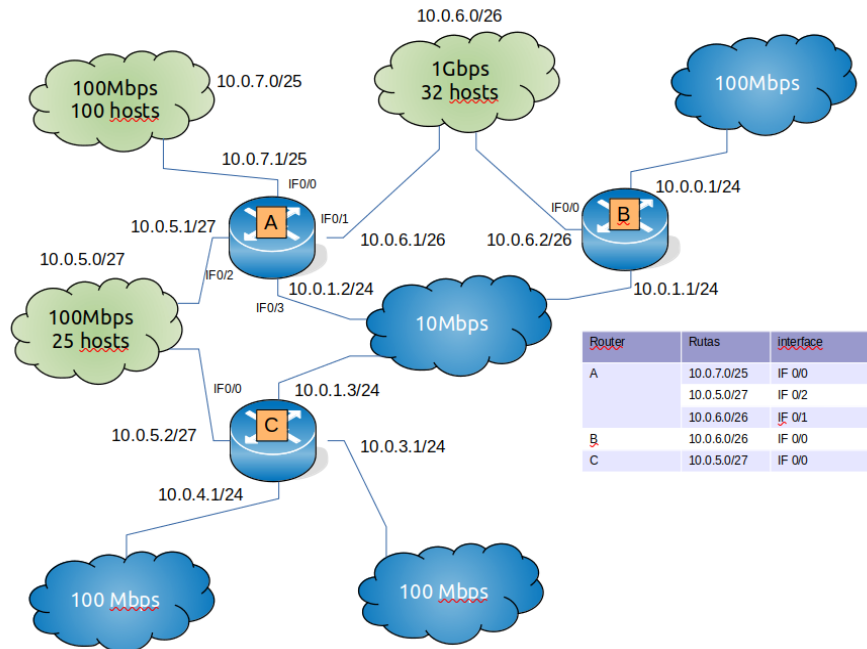
En la figura se muestra una topología de red, que en principio solo cuenta con los routers A, B y C y las redes azules (las que no indican la cantidad de hosts). Todo se encuentra configurado correctamente, de manera que cualquier host de cualquier red pueda comunicarse con cualquier otro.

Se planea agregar las 3 redes verdes (las que indican la cantidad de hosts) y conectarlas como se indica en la figura. Se cuenta con 3 rangos de direcciones para ser asignados a cada una de las redes: 10.0.5.0/24, 10.0.6.0/24 y 10.0.7.0/24.



- Asigne un rango **mínimo** de direcciones a cada red y detalle su dirección de broadcast y de red. Además liste las entradas que habría que agregar a los routers para que dichas redes tengan conectividad con el resto. De ser necesario, nomencle arbitrariamente las interfaces de los routers.

Rta:



- b. La red usa OSPF como protocolo de ruteo. Mostrar un posible frame OSPF generado por A, una vez que convergieron la rutas. Como métrica usar  $10^9/\text{ancho\_de\_banda}$ . Además muestre la tabla de forwarding completa del router A mostrando sólo las rutas con mejores métricas.

Use el modelo de mensaje OSPF que se encuentra en el encabezado de la práctica de ruteo

Rta:

El router A hace un flood con la información de las redes directamente conectadas

|                 |
|-----------------|
| ID: A           |
| SEQ: 1          |
| TTL: 4          |
| 10.0.7.0        |
| 255.255.255.128 |
| 10              |
| 10.0.6.0        |
| 255.255.255.192 |
| 1               |
| 10.0.5.0        |
| 255.255.255.224 |
| 10              |
| 10.0.1.0        |
| 255.255.255.0   |
| 100             |

| Network     | Next Hop |
|-------------|----------|
| 10.0.7.0/25 | IF 0/0   |
| 10.0.6.0/26 | IF 0/1   |
| 10.0.5.0/27 | IF 0/2   |
| 10.0.1.0/24 | IF 0/3   |
| 10.0.3.0/24 | 10.0.5.2 |
| 10.0.4.0/24 | 10.0.5.2 |
| 10.0.0.0/24 | 10.0.6.2 |

## Ejercicio 4

- a. Dada la siguiente tabla describiendo el comportamiento de algunas variables en el transcurso de una conexión TCP.

| RTT | CWND | FlightSize | LBS  |
|-----|------|------------|------|
| 1   | 60KB | 30KB       | 30KB |
| 2   | 60KB | 60KB       | 60KB |
| 3   | 2KB  | 2KB        | 2KB  |
| 4   | 4KB  | 4KB        | 6KB  |
| 5   | 8KB  | 8KB        | 14KB |
| 6   | 16KB | 16KB       | 30KB |
| 7   | 30KB | 30KB       | 60KB |

- Describa qué puede suceder en una conexión para que *CWND* y *FlightSize* se comporten como en los RTT 1, 2 y 3.
- Ídem para los RTT 6 y 7.

Rta: (hay otras explicaciones válidas también)

- En el primer RTT se envían 30 kB y no se reciben ACKs reconociendo nuevos datos. En el segundo RTT, se envían otros 30 kB y no se reciben ACKs reconociendo nuevos datos. En el tercer RTT, vence el RTO y se empieza con LW = 1 SMSS, con SS. Ssthresh va a 30 kB
  - En 6 se sigue con SS y se mandan 16 KB, vuelven los 8 ACKs. En el 7, se usan 7 ACKs para crecer según SS, CWND += SMSS por ACK. Entonces CWND queda en 16 kB + 14 kB = 30 kB. El último ACK se usa para crecer según CA. CWND += 2 kB \* 2 kB / 30 kB. Este incremento marginal se puede redondear a 0 bytes en este RTT.
- b. Luego del envío de datos, el host emisor y el receptor envían simultáneamente un segmento sin datos que tiene sólo el flag FIN prendido. Describir una secuencia válida de envío de segmentos y cambios de estados en ambos extremos de la conexión hasta que se termina de cerrar.

Rta:

Ambos reciben el segmento con FIN y pasan a FIN\_WAIT\_1.

Ambos envían un ACK y pasan a TIME\_WAIT

Después de 1 minuto (2 life time segments), ambos pasan a CLOSED.

## Ejercicio 5

En la casilla de mail de un usuario llamado Dilbert se encuentra el siguiente mail:

```
To: dilbert@memento.com
From: devil564@hotmail.com
Subject: La salvación está aquí!
MIME-Version: 1.0
Content-Type: text/html; charset = "iso-8859-1"
<html> <head></head>
<body>
<iframe src="http://bewithus.com" /><br />
YA estás salvado hermano! Nuestros escritos proveen la salvación:
<br />
<a href="http://salvation.com/newmember.php?level=acolino">Click para salvación!</a><br />
<br />

</body>
</html>
```

Suponiendo que en su computadora el usuario tiene un *user agent* configurado con un servidor POP3 (pop3.memento.com.ar) y además un *web proxy* (proxy.memento.com.ar):

- a. Describa los flujos de datos (TCP o UDP) que se desencadenan cuando el usuario decide bajar a su computadora el mail describiendo los hosts involucrados.

Rta:

```
< host, resolver, *, 53, UDP> x2
< host, pop3.memento.com.ar, *, 110, TCP>
```

```
< host, proxy.memento.com.ar, *, 80, TCP>  
< proxy.memento.com.ar, resolver, *, 53, UDP> x2  
< proxy.memento.com.ar, salvation.com, *, 80, TCP>  
< proxy.memento.com.ar, bewithus.com, *, 80, TCP>
```

b. Detalle todos los mensajes DNS y HTTP que tienen como cliente al proxy web.

Rta:

DNS:

- 1) Se envía un DNS query (A, bewithus.com) a resolver
- 2) Resolver responde con dirección IP de bewithus.com
- 3) Se envía un DNS query (A, salvation.com) a resolver
- 4) Resolver responde con dirección IP de salvation.com

HTTP:

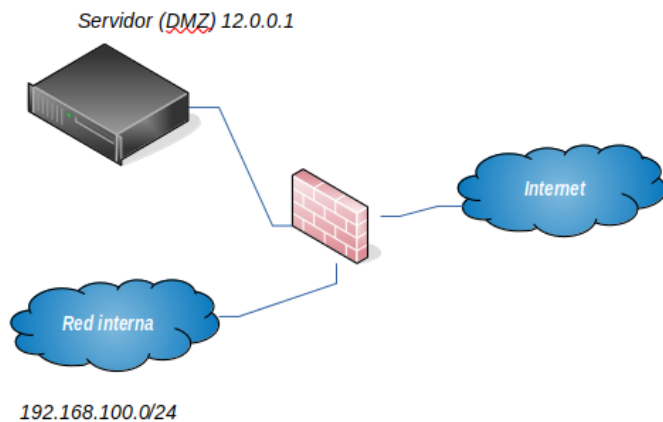
- 1) Se envía un GET pidiendo el html como el siguiente: GET / HTTP/1.1 Host: bewithus.com y vuelve 200
- 2) Se envía un GET pidiendo la imagen como el siguiente: GET /panlfleto1.png HTTP/1.1 Host: salvation.com y vuelve 200
- 3) Se envía un GET pidiendo la imagen como el siguiente: GET /panlfleto2.png HTTP/1.1 Host: salvation.com y vuelve 200

Asumir(para todo el ejercicio):

- La PC del usuario y el web proxy conocen la dirección IP del Resolver DNS de su proveedor de internet.
- Todos los servicios HTTP involucrados son HTTP/1.1
- Las caches de los servicios involucrados se encuentran vacías al iniciar el proceso.
- Para el análisis, ignore las consultas DNS que podría originar el Resolver del proveedor de internet

## Ejercicio 6

La red de una universidad se muestra en la figura. En la misma, se usa una DMZ de manera de exponer una máquina que posee el servidor web y el servidor de mail para el dominio asignado a la universidad. Además, este server actúa como resolver DNS y es autoritativo para el dominio de la universidad. Desde la red interna, los usuarios pueden navegar la web de forma segura y no segura, y realizar consultas DNS recursivas al resolver y enviar y recibir mails por medio del servidor (el servidor posee casillas accesibles con POP3). El servidor DNS debe poder contestar *queries* hechas desde Internet. La dirección IP del server es 12.0.0.1 y la red interna tiene el rango 192.168.100.0/24



- a. Muestre las reglas de *firewall statefull* que permitan proteger a la red de la universidad de posibles atacantes en Internet

Rta:

Regla default: DROP

DNS:

Consultas internas (recursivas):

$\langle 192.168.100.0/24, *, 12.0.0.1, 53, UDP \rangle$

Consultas resolver (iterativas):

$\langle 12.0.0.1, *, Internet, 53, UDP \rangle$

Consultas externas:

$\langle Internet, *, 12.0.0.1, 53, UDP \rangle$

Web:

web y web segura:

$\langle 192.168.100.0/24, *, Internet, 80, TCP \rangle$

$\langle 192.168.100.0/24, *, Internet, 443, TCP \rangle$

web server:

$\langle Internet, *, 12.0.0.1, 80, TCP \rangle$

$\langle 192.168.100.0/24, *, 12.0.0.1, 80, TCP \rangle$

Mail:

Una para que se puedan recibir mails desde Internet:

$\langle Internet, *, 12.0.0.1, 25, TCP \rangle$

Una para que se puedan enviar mails a Internet:

$\langle 12.0.0.1, *, Internet, 25, TCP \rangle$

Una para que se puedan descargar mails a la red interna:

$\langle 192.168.100.0/24, *, 12.0.0.1, 110, TCP \rangle$

Una para que se puedan enviar mails desde la red interna:

$\langle 192.168.100.0/24, *, 12.0.0.1, 25, TCP \rangle$

- b. Los usuarios necesitan poder garantizar la autenticidad del servidor de mails y para esto usan una conexión segura TLS. Explique dónde deben instalarse el/los certificado/s digitales para que los usuarios puedan garantizar la autenticidad del servidor de mails. Indique además cómo se valida ese certificado.

Rta:

Un certificado firmado por una CA debe instalarse en el servidor de mails. Este envía el certificado durante el handshake TLS y los usuarios lo validan con la clave pública de la CA que deberían tener instalado en sus computadoras.